UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/686,882	10/15/2003	Kenneth Mark Williams	PA2594US	4103

22830 7590 09/26/2006

CARR & FERRELL LLP
2200 GENG ROAD
PALO ALTO, CA 94303

EXAMINER	
THOMAS, SHANE M	

ART UNIT	PAPER NUMBER
2186	

DATE MAILED: 09/26/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/686,882	WILLIAMS ET AL.	
	Examiner	Art Unit	
	Shane M. Thomas	2186	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 July 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-23* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *15 October 2003 and 10 July 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *7/10/06 and 7/24/06*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This Office action is responsive to the amendment filed 7/10/2006. Claims 1-23 are pending. Applicants' arguments and amendments to the claims have been carefully considered, but they are not persuasive and do not place the claims in condition for allowance. Accordingly, this action has been made FINAL.

Excerpts from all prior art references cited in this Office action shall use the shorthand notation of [column # / lines A-B] to denote the location of a specific citation. For example, a citation present on column 2, lines 1-6, of a reference shall herein be denoted as "[2/1-6]."

All previously outstanding objections and rejections to the Applicant's disclosure and claims not contained in this Action have been respectfully withdrawn by the Examiner hereto.

Response to Amendment

Examiner has cited the prior art reference of Ramapgopal et al. (¶¶31-35) to teach the amended limitations of independent claims 1,8,15,16,17, and 21. Updated claim rejections are discussed below.

Information Disclosure Statement

The IDSs filed 7/10/2006 and 7/24/2006 have been considered by the Examiner. Signed copies have been attached herewith.

Response to Arguments

Applicant's arguments filed 7/10/2006 have been fully considered but they are not persuasive.

As per the argument regarding claims 5 and 12 (page 13 of the response), Applicant argues "the processing of sequences of arbitrary length occurs in the data aligner, ... in the peripheral device 15 that is separate from the host processor 10." While the Petersen does suggest that the processing of data sequences occurs in the peripheral device 15, the scope of claim 10, as presently drafted by the Applicant, merely claims a computing comprising a general-purpose processor, while making no correlation to the function of that general purpose processor. Petersen clearly shows a general-purpose processor 10 of the computing system depicted in figure 7 and thus clearly teaches the claim limitation.

As per the argument regarding claims 4 and 11 (pages 14-15 of the response), in response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, Applicant argues, "because the operations of the peripheral device for data alignment in Petersen is unrelated to the extended instruction processor in Emma, one skilled in the art would not have combined the references." The Examiner respectfully disagrees. The scope of the arguments of the Applicant do not match the scope of the claimed invention. Irregardless if the data aligner is

not being used directly by the host processor 10 of Peterson, claim 4 only claims the computer system comprising a processor having an extensible instruction set and makes no correlation between that processor and the operation of the load/store buffer. An extensible processor instruction set has many benefits as discussed in Emma, and while the scope of the invention of Peterson may not directly entail the specific processes and procedures executed by the processor 10, it is sufficient to assume (as would have been recognized by one of ordinary skill in the art) and necessarily inherent that the processor 10 would have executed instructions, as is the general function of a system processor. Emma teaches a method to circumvent compatibility issues between legacy software (i.e. a current instruction set as implemented by the host processor 10 of Petersen) and an architectural extension (i.e. instruction extensions not currently used but may be needed to one of ordinary skill using the system of Petersen and looking to upgrade the processing system of figure 7 with newer instructions). It would have been obvious to that one of ordinary skill to combine Petersen and Emma as Emma teaches a means to allows for the adding of the new instructions without recompilation and reassembly of Petersen, thereby saving the user time and complexity when integrating newer instructions into the instruction set of Petersen.

Claim Objections

Claims 13 is objected to because of the following informalities:

As per claim 13, the term --an next aligned word-- should be amended to --***a*** next aligned word--.

Appropriate correction is required.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-3,5,6,8-10,12,13, and 15-23 are rejected under 35 U.S.C. 103(a) as being unpatentable by Petersen (U.S. Patent No. 5,517,627) in view of Ramagopal et al. (U.S. Patent Application Publication No. 2003/0196058). Further the prior art reference of Voith et al. (U.S. Patent No. 5,636,224) is being used to support an inherent feature of the Petersen reference.

As per claims 1,15, and 17, Petersen teaches a **method** (figures 10 and 13), **system** (figure 8), and **processor** 10 **configured to execute GET** (read) and **PUT** (write) **instructions** (write and read aligners can be implemented with a ROM, which is well known in the art to contain instructions for controlling hardware systems, - [13/7-11]) for **processing data sequences of arbitrary length in a computing system**. Alternatively, the Examiner is considering the memory controller 26 as being a --processor-- as the memory controller can be seen in regard to figure 8 of Petersen to interact with the buffer memory 27 and the load/store buffer 24. Since the instructions to load/store the buffer 24 may be implemented via a ROM [13/8-11], it is necessarily inherent that some element perform the operation of executing the instructions stored on the ROM in order to utilize the load/store buffer 24. That element is being considered by the Examiner to be the --processor--.

Petersen teaches **initializing a load/store buffer** (data aligner 24, figure 8) **by loading a first aligned word of fixed length** (4 bytes - [7/55-63]) **into the load/store buffer** (figure 5A -

[8/41-47]) **from memory** 27 - [12/18-20]. The Examiner is considering loading a data word from the buffer 27 to the data aligner 24 as --initializing-- since the process places data into the data aligner. It can be seen with respect to figures 5A and 5B that it takes two loads to the data aligner 24 from the buffer memory 27 to --initialize-- the data aligner by placing data therein.

Further, Petersen teaches **further initializing the load store buffer** 24 **by loading a second aligned word** (figure 5B) **into the load/store buffer** [8/48-50] **from the memory, reading one or more data sequences** (wherein a single data sequence is being defined by the Examiner as the number of bits transferred per lane L2(i), in the present embodiment of Petersen - one byte (8 bits) - [7/55-63]) **from the load/store buffer, such that the total length of the sequences in each read does not exceed the fixed length of the first aligned word** ([8/30-58] which states that a full-width read is 4 bytes - [8/56-57]), **and loading additional aligned words to the load/store buffer from the memory** 27 **to replace data sequences that are read** ([13/51/54], where the reading process can be repeated as necessary to read data from the buffer memory 27 - [figure 13, step 153 - step 142]).

Peterson teaches that a DMA controller 13 is used to issue read and write requests on the host bus 14 to the peripheral device 15, thereby alleviating host processor from performing such duties [9/49 - 10/2], in addition to sending data from the system memory 12 to the peripheral device 15 [9/62-66]. The data retrieved or written to the buffer 27 is sent to the host bus 14 - [12/18-20]. However, while Peterson teaches reading one or more data sequences from the load/store buffer 24, the reference does not specifically teach reading those data sequences **into a register file for instruction execution**. Register files are well known in the processing and

memory arts to be the collection of local processor registers (see noted prior art of record but not relied upon in the Conclusion section of this action).

Ramagopal teaches in ¶¶30-31 that data is received and sent between the memory 12 and register file 28 for instruction execution and storage after execution instruction, respectively (figure 1). Ramagopal further teaches that DMA accesses may be used to access memory 12 to store data and the like between the memory 12 and an external source (¶35). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have combined the load/store buffering unit of Petersen with the teaching of associating a DMA access (DMA to/from external source to/from a memory that interfaces the register file) with a register file of Ramagopal in order to have (1) enabled the processor 10 of Petersen to have buffered the data to be executed in a memory (register file) very close in relation to the processor (to avoid having to continually fetch the data for each execution from the memory 12 during every instruction execution, thereby increasing instruction throughput) and (2) utilized the DMA processing to quickly send and receive data between the memory 12 and an external source (for example the peripheral device 15 of Petersen) for forwarding on to the register file - ¶35 of Ramagopal.

As a result of the combination of Petersen and Ramagopal, it can be seen that the **data sequences** from the **load/store buffer** 24 of modified Petersen would have been **read into the register file** (through the DMA controller's interaction between the memory 12 via the host bus 14 and the external device 15, comprising the load/store buffer 24), where the data sequences would have been used for **instruction execution** - ¶35 of Ramagopal.

As per claim 2, Petersen teaches that **the data sequence length** [read] **is a byte** [8/42-43] and [7/60-61].

As per claim 3, Petersen teaches that **the data sequence length** [read] **is a bit** [7/64-67].

As per claims 5 and 12, Petersen teaches **wherein the computing system comprises a general-purpose processor** [host processor 10] - figure 7. The Examiner is considering the entirety of figure 7 of Petersen to be a --computing system--, which thereby includes the detailed view of the peripheral device of figure 8.

As per claims 6 and 13, Petersen inherently teaches **wherein the first aligned is stored in a first memory location and the second [next] aligned word is stored in an adjacent second memory location** (the aligned words are stored in a FIFO [10/12], so as well known in the art and an inherent feature of Petersen, a FIFO buffer stores a subsequent piece of data in an adjacent location to the location of the previous piece of data - refer to figure 3 of Voith), **and the second memory location is accessed by incrementing a memory address pointer after the first aligned word** (a further inherent feature of a FIFO, since a write pointer must be incremented to the next memory location after data has been written to a present FIFO memory location - refer to Voith figure 4 and [5/63-66]).

As per claims 8,16, and 21, the rejection follows similarly to the rejection of claims 1,15, and 17. Petersen teaches **initializing a load/store buffer** (data aligner 24) **with one or more unaligned data sequences** [4/22-61], **such that the total length of each data sequence does not exceed the fixed length of an aligned word** (i.e. 32-bit word) [6/53-7/25] and figures 4A-4C. It should be noted that the Examiner is consider the step of write initializing to be the step of simply writing a first data sequence to the data aligner 24, such as the writing of data **01** in the

data aligner 24 (as shown being queued in figure 4A - [6/53-55]. Further, Petersen teaches **writing one or more unaligned data sequences to the initialized load/store buffer** 24, **such that the written unaligned data shifts an aligned word into a memory location** (into the buffer 27) - figure 4B and [6/59-63]. Finally, Petersen teaches **flushing of the load/store buffer** 24 **in order to store any of the remaining unaligned data in order to store any of the remaining unaligned data into memory** (figure 4C and [6/64-7/4]). Figure 4C and [6/64-7/4] teaches that the unaligned data 01,02,03, and 04, now configured to an aligned word, is flushed (i.e. sent to the buffer 27 as taught above) while remaining unaligned data 05 is stored into memory (i.e. the queue(s) of the data aligner - [7/1-4]).

Peterson teaches that a DMA controller 13 is used to issue read and write requests on the host bus 14 to the peripheral device 15, thereby alleviating host processor from performing such duties [9/49 - 10/2], in addition to sending data from the system memory 12 to the peripheral device 15 [9/62-66]. The data retrieved or written to the buffer 27 is sent to the host bus 14 - [12/18-20]. However, while Peterson teaches initializing the load/store buffer 24 by filling the load/store buffer with one or more unaligned data sequences, the reference does not specifically teach data sequences being acquired from **a register file for** [used for] **instruction execution**. Register files are well known in the processing and memory arts to be the collection of local processor registers (see noted prior art of record but not relied upon in the Conclusion section of this action).

Ramagopal teaches in ¶¶30-31 that data is received and sent between the memory 12 and register file 28 for instruction execution and storage after execution instruction, respectively (figure 1). Ramagopal further teaches that DMA accesses may be used to access memory 12 to

store data and the like between the memory 12 and an external source (¶35). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have combined the load/store buffering unit of Petersen with the teaching of associating a DMA access (DMA to/from external source to/from a memory that interfaces the register file) with a register file of Ramagopal in order to have (1) enabled the processor 10 of Petersen to have buffered the data to be executed in a memory (register file) very close in relation to the processor (to avoid having to continually fetch the data for each execution from the memory 12 during every instruction execution, thereby increasing instruction throughput) and (2) utilized the DMA processing to quickly send and receive data between the memory 12 and an external source (for example the peripheral device 15 of Petersen) for forwarding on to the register file - ¶35 of Ramagopal.

As taught in ¶31 of Ramagopal, after instruction execution data can be sent to memory 12, where the data can further be sent to the peripheral device 15 containing the load/store buffer 24 (of modified Petereson) via a DMA transaction - ¶35 of Ramagopal.

As per claim 9, Petersen teaches that **the data sequence length** [written] **is a byte** - figure 4A, [6/53-59] and [4/7-10].

As per claim 10, Petersen teaches that **the data sequence length** [written] **is a bit** [4/9-10].

As per claim 18, Petersen teaches that **the number of data sequences read is an immediate specified number** that is specified by the host [8/62-67].

As per claim 19, Petersen teaches that **the number of data sequences read is a specified number stored as an index in a register memory** [9/4-6]. The Examiner is considering the value of the "CURRENT READ" to be an index.

As per claim 20, Petersen teaches **in which a first one of the one or most data sequences read is located at a first memory location** (i.e. any of the bytes of the 4 byte word entry of FIFO buffer 27 that is first read into the data aligner upon a read request from the host - [8/41-46] that initializes the data aligner as discussed supra) **and the one or more data sequences comprises a specified number of data sequences stored in a register memory** 27 (here the Examiner is considering the buffer memory 27 as being the register memory and the --specified number-- of data sequences stored at the first index of the [FIFO] buffer 27 is specified to be four - as each [FIFO] buffer 27 entry contains four data sequences (four aligned bytes) - [1/26-35] and [8/30-50] - where the --first index-- is the index of the first entry of the FIFO buffer 27 to be read), **wherein the subsequent data sequence following the first of the data sequences is located at a second memory location pointed to by a second index** (the next memory location that is accessed from the FIFO buffer 27 - [8/48-50] as taught in the following example).

Petersen teaches that the first data sequence read can be any of the data sequences of the first FIFO buffer, with data unit 01 being used as an example that is read first [8/41-47]. If for example, the 04 data sequence was read first (i.e. the --first of the one or more data sequences--), thereby leaving data sequences 01,02, and 03, in the queues of the data aligner, data sequences 01, 02, 03, 05, 06, 07, and 08, would have been available to the host [8/51-52]. Thus it can be seen that the **subsequent data sequence** (data sequence 05) **following the first of the data**

sequences (04) **is located at a second memory location** (buffer 27, second entry) that is **pointed to by a second index** (i.e. whatever the index/address of the FIFO buffer 27 contains the next data word that is read after the first data word - [8/48-50]).

As per claim 22, Petersen teaches **the number of unaligned data sequences written is an immediate specified number** that is specified by the host [5/41-46], in similar fashion to the number of requested data sequences read [8/62-67].

As per claim 23, Petersen teaches that **the number of unaligned data sequences written is a specified number stored at an index in a register memory** where the --register memory-- is the combination of the registers S1(0) through S1(N) [6/16-20] with each register indicating that an unaligned data sequence is stored in the respective queue. In other words the specified number is an index since the registers S1(0) to S1(N) need to be indexed to acquire the number of unaligned data sequences that are queued in the data aligner 24 from a previous write request; these indexed registers (collectively a --register memory--) combine to produce the CURRENT QUEUED value.

Claims 4 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Petersen (U.S. Patent No. 5,517,627) in view of Ramagopal et al. (U.S. Patent Application Publication No. 2003/0196058), as applied to claims 1-3,5,6,8-10,12,13, and 15-23 above, in further view of Emma (U.S. Patent No. 5,619,665).

As per claims 4 and 11, Petersen teaches a general-purpose processor (host processor 10 - figure 7) but does not specifically teach a **processor having an extensible instruction set**. Emma teaches in the abstract that extension of an instruction set allows for circumventing

software compatibility issues when allowing legacy software to benefit from new architectural extensions without recompilation and reassembly. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have combined the data aligner system of figure 7 of Petersen with the teaching of an extension instruction set of Emma in order to have gained expandability of the instruction set of the processor 10 of Petersen, thereby gaining flexibility and compatibility for future instruction extensions without having to recompile or reassemble the system of Petersen.

Claims 7 and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Petersen (U.S. Patent No. 5,517,627) in view of Ramagopal et al. (U.S. Patent Application Publication No. 2003/0196058), as applied to claims 1-3,5,6,8-10,12,13, and 15-23 above, in further view of Voith et al. (U.S. Patent No. 5,636,224).

As per claims 7 and 14, the rejections of lines 1-2 follows the rejection of lines 1-2 of claims 6 and 13, respectively. Petersen (with Voith as a supporting reference) teaches that the claimed invention inherently uses incrementing of an address pointer to access a second memory location instead of decrementing the address pointer to access the second memory location. Voith teaches **that the second memory location is accessed by decrementing a memory address pointer after the first aligned word is accessed** [6/35-43] is an equivalent method known in the art of FIFO accessing. Therefore, because these two method of accessing a subsequent memory location in a FIFO buffer were art-recognized equivalents at the time the invention was made, one of ordinary still in the art would have found it obvious to substitute

decrementing a memory address pointer for incrementing a memory address pointer of a FIFO buffer.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Hennessy et al. teaches a register file as being part of a processor on page 345.

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane M. Thomas whose telephone number is (571) 272-4188. The examiner can normally be reached M-F 8:30 - 5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matt M. Kim can be reached at (571) 272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane M. Thomas

PIERRE BATAILLE
PRIMARY EXAMINER
9/20/06